FRE Version History

Amy Langenhorst <Amy.Langenhorst@noaa.gov>

This document describes the version history of FRE, the FMS Runtime Environment. For usage information, see http://cobweb.gfdl.noaa.gov/~arl/fre.

Table of Contents

1. fremake Version History

1.1. New in fremake01

1. new version of mkmf allows for calling mkmf from exec dir instead of src dir. fremake uses this to allow you to compile the optimized and debug versions at the same time.
2. new option -p to call list_paths only once, after all the cvs stuff is complete. This is much faster. This works for all experiments except mom4 ones that treat the mom4 paths differently than the shared code paths.

2. frerun Version History

2.1. New in frerun01

1. namelists in runscript will be sorted alphabetically

2. new attribute zetac_layout is processed
3. modified subroutine getexecutable to always check whether an inherited experiment should have its own executable or not
4. bugfix to production runs of less than 1 year; will now set an appropriate combineFreq for these runs
5. new option for reading all the namelists for an experiment and outputting all namelists alphabetically as text in xml tags.
6. bugfix to procedure of finding the debug executable for inherited experiments
7. changed so that project tag gets placed in regression runs as well as production runs

2.2. New in rtsrun14

1. ability to run on altix platform

2.3. New in rtsrun13

1. prints a warning if make_exchange_reproduce is false for regression tests, and prints a warning if make_exchange_reproduce is true for production runs.
2. added code to combine multithreaded write restart files
3. added code to change priority projects to dev projects when submitting a windf job

2.4. New in rtsrun11, rtsrun12

1. just call the corresponding version of the postprocessing program

2.5. New in rtsrun10

1. better wording on namelist warnings
2. supports platform csh for changing version of compiler. To use the new version of the compiler, use xml as follows:

```
<setup>
   <target platform="sgi">
      <csh>
        source /opt/modules/modules/init/csh
        module switch mpt mpt_1900
        module switch mipspro mipspro_741m
      </csh>
   </target>
</setup>
```

   and remove the old <targetPlatform>sgi</targetPlatform>.
3. multiple types of input csh supported

```
<input>
  <csh> this is executed in the section which copies input files at the beginning
        of each JOB, same as previously
  </csh
  <csh type="init"> this is executed in the section which copies input files ONLY ONC
        PER EXPERIMENT, only at the initial run time t=0
  </csh>
  <csh type="always"> this is executed in the mpirun loop so that it is executed
        BEFORE EVERY MPIRUN command
  </csh>
```

2.6. New in rtsrun9

1. developmental changes, calls rtspp9

2.7. New in rtsrun8

1. Doesn't attempt to compile an executable if you specified an existing one in your xml
2. can now set project for jobs with rtspriority

3. frepp Version History

3.1. New in frepp01

1. now using Fanrong's Analysis.pm package for figures
2. modified not to cpio files when there is only one file
3. prevent trying to run platform csh intended for the altix on the sgi platform
4. added the ability to average daily data to a monthly timeseries. To take advantage of this capability, you need to add three lines to your xml:

```
        <component type="atmos" zInterp="ncep" start="0001" source="atmos_month">
           <timeSeries freq="daily" source="atmos_daily" chunkLength="5yr"/>
           <timeSeries freq="monthly" chunkLength="5yr"/>
           ...
           <timeSeries freq="monthly" averageOf="daily" chunkLength="5yr">
              <variables> t_ref_min t_ref_max </variables>
           </timeSeries>
        </component>
```

The new feature is 'timeSeries freq="monthly" averageOf="daily"'. There are several things to note:

a. You can either calculate this in the same pp job with the 'timeSeries freq="daily"' or do it anytime after

the daily timeSeries has been created. (ie, you can run just this piece of postprocessing offline.)

b. The chunkLength requested for the averaged file must be the same as the chunkLength of the daily timeSeries file, ie, 5yrs. Longer chunks of pp should pick up the new variables and create 20yr timeSeries from the 5yr ones.

c. This is specifically designed with the noleap calendar in mind and currently does not check the model calendar.

d. The variables tag is required for this type of timeSeries calculation.

5. gory bugfixy details surrounding making cpio files, calculating one or two variables offline after other variables have been processed, and performing frescrub at any point along the way.
6. the variables tag is now available for all timeSeries options.
7. checks that timeSeries contain the right number of time levels as it calculates them. If the timeSeries file contains the wrong number of levels, you'll get email about it.

3.2. New in rtspp14

1. ability to run on altix platform
2. analysis figure improvements

3.3. New in rtspp13

1. analysis figure bugfix when using <variables> tag inside of <timeSeries>; previously rtspp could mistakenly assume that pp files were missing and not create figures for that data
2. new timeSeries and timeAverage attribute 'from' can override what chunklength or interval pp you want to calulate a timeSeries or timeAverage from, ie

```
<timeSeries freq="monthly" chunkLength="200yr" from="100yr"/>
```

3. annual multi-year averages will only try to calculate the interval they're supposed to calculate. Previously, they looked for the last existing output file and calculated all the intervals after that through the -t year. This change is only useful when you are trying to run offline pp out of order.
4. seasonal averages that need the previous december now know how to get it from the history file as well as from the previously existing monthly timeAverage file. This change is only useful when you are trying to run offline pp out of order.
5. better error message when you try to run seasonal timeSeries without any available monthly timeSeries data
6. -m option: If you give the -m option, you will get an email when your job is complete. Only useful for offline pp. Doesn't send email if option -p is used; this way you only get email when it's done with the stream
7. can put <variables> inside seasonal ts and it will use only those variables. Previously you could only use <variables> in the monthly ts, and those variables would then get used for seasonal and annual ts also. Note that you still need to have the monthly ts for all variables you request a seasonal ts of.
8. minor cleanup: seasonal ts tests whether cpio has already been extracted

3.4. New in rtspp12

1. BUGFIX: timavg.csh was using the wrong weighting interval in some cases
2. truncation of long filenames on timavg'ed files and plevel data avoided
3. bugfix for postprocessing jobs run after an experiment has been frescrubbed
4. dmget improvements to avoid the 'line too long' error

3.5. New in rtspp11

1. doesn't call nc_to_midtime, should improve speed of postprocessing by 10-25%
2. consolidates error notification into fewer actual emails
3. doublechecks that output directories exist so that there are no missing directories
4. additional dmgets implemented
5. static files created with splitncvars, should retain all appropriate attributes
6. time_bounds bugfix to seasonal averages when the first december of the run needs to be recalculated from the end of the first year
7. new analysis figure options:

```
          -Y year      = specify a four digit year as your analysis's start year, ex
                         This year overrides the startYear specifed in the <analysis>
          -Z year      = specify a four digit year as your analysis's end year, ex -Z
                         This year overrides the endYear specifed in the <analysis> t
```

3.6. New in rtspp10

1. analysis figures fixes
2. time_bounds fixes
3. bugfixes and improvements to external utilities:list_ncvars, timavg, plevel

3.7. New in rtspp9

1. developmental changes for time_bounds fixes

3.8. New in rtspp8

1. added option to split postprocessing into different jobs: -c component
   - without -c, behavior is same as rtspp6
   - keyword '-c split' means each component will be managed in a separate postprocessing stream; ie, one job will be submitted for each component.
   - using a component name, ie '-c ocean', means that only the postprocessing for that component will be

done.

2. ncrcat now uses -O option to overwrite files if they already exist. Previously if you were regenerating pp after a system error, you would get emails about ncrcat errors. ncrcat complained because the output file already existed. This change will prevent those error messages.
3. includes seasonal average functionality.
4. uses ncatted to put the calendar attribute back after plevel.sh called
5. filename attribute corrected in some annual average files
6. writes some version information to ppRootDir/.frepp_history